```
MODULE  RdcFoo;                                    (*  Date   :  21  July  2012     *)
                                                   (*  Author :  Robert D Campbell  *)


IMPORT  Dialog, Files, Stores;


CONST
  status      =  'FooStatus';
  minVersion  =  0;
  maxVersion  =  2;
  magic       =  170702012;


TYPE
  Dlg  =  RECORD
            a*, b*, c*  :  REAL;
            p*, q*, r*  :  INTEGER;
          END;


VAR
  dlg*  :  Dlg;

(**  *****  *****  *****  *****  Commands  *****  *****  *****  *****  **)

PROCEDURE  Do*;
  BEGIN
  END  Do;


(**  *****  *****  *****  *****  Status  *****  *****  *****  *****  **)

PROCEDURE  ReadStatus;
  VAR
    loc           :  Files.Locator;
    file          :  Files.File;
    rd            :  Stores.Reader;
    thisV, hocus  :  INTEGER;
  BEGIN
    loc  :=  Files.dir.This ('').This ('Rdc/Rsrc');
    IF  loc  #  NIL  THEN
      file  :=  Files.dir.Old (loc, status + '.bin', FALSE);
      IF  file  #  NIL  THEN
        rd.ConnectTo (file);
        rd.ReadInt (thisV);
        ASSERT (thisV  <=  maxVersion, 44);  ASSERT (minVersion  <=  thisV, 45);

        rd.ReadReal (dlg.a);
        rd.ReadReal (dlg.b);
        rd.ReadReal (dlg.c);

        rd.ReadInt (dlg.p);
        rd.ReadInt (dlg.q);
        IF  thisV  >=  1  THEN  rd.ReadInt (dlg.r)  ELSE  dlg.r  :=  3  END;

        rd.ReadInt  (hocus); ASSERT (hocus = magic, 45);
        file.Close
      END
    END
  END  ReadStatus;


PROCEDURE  SaveStatus;
  VAR
    loc   :  Files.Locator;
    file  :  Files.File;
    wr    :  Stores.Writer;
    k     :  INTEGER;
  BEGIN
    loc  :=  Files.dir.This ('').This ('Rdc/Rsrc');
```

```
    IF  loc  =  NIL  THEN  RETURN  END;
    file  :=  Files.dir.New (loc, Files.ask);
    IF  file  #  NIL  THEN
      wr.ConnectTo (file); wr.WriteInt (maxVersion);

      wr.WriteReal (dlg.a);
      wr.WriteReal (dlg.b);
      wr.WriteReal (dlg.c);

      wr.WriteInt (dlg.p);
      wr.WriteInt (dlg.q);
      wr.WriteInt (dlg.r);

      wr.WriteInt  (magic);
      file.Register (status, 'bin', Files.ask, k)
    END
  END  SaveStatus;


BEGIN
  dlg.a  :=  3.141592653589793;                    (*  Default values  *)
  dlg.b  :=  0.577215664901533;
  dlg.c  :=  2.718281828459045;

  dlg.p  :=  1;
  dlg.q  :=  2;
  dlg.r  :=  3;

  ReadStatus;

  Dialog.Update (dlg)
CLOSE
  SaveStatus
END  RdcFoo.
```

❗ DevDebug.Unload

❗ "StdCmds.OpenToolDialog ('Rdc/Rsrc/Foo', '(Rdc)Foo')"

```
PROCEDURE  ExternalizeFields* (ptr : ANYPTR; mask : SET; VAR wr : Stores.Writer);
  VAR
    item  :  Meta.Item;
    sc    :  Meta.Scanner;
    cnt   :  SHORTINT;
  BEGIN
    cnt  :=  0;
    wr.WriteSet (mask);
    Meta.GetItem (ptr, item);
    sc.ConnectTo (item);
    sc.Scan;
    WHILE  ~sc.eos  DO
      IF  (sc.this.typ  IN  mask)  &  (sc.this.vis  =  Meta.exported)  THEN
        CASE  sc.this.typ  OF
          Meta.boolTyp  :  wr.WriteBool (sc.this.BoolVal ()); INC (cnt)
   |      Meta.charTyp  :  wr.WriteChar (sc.this.CharVal ()); INC (cnt, 2)
   |      Meta.intTyp   :  wr.WriteInt  (sc.this.IntVal  ()); INC (cnt, 4)
   |      Meta.longTyp  :  wr.WriteLong (sc.this.LongVal ()); INC (cnt, 8)
   |      Meta.realTyp  :  wr.WriteReal (sc.this.RealVal ()); INC (cnt, 8)
   |      Meta.setTyp   :  wr.WriteSet  (sc.this.SetVal  ()); INC (cnt, 4)
        ELSE
          HALT (20)
        END
      END;
      sc.Scan
    END;
    wr.WriteSInt (cnt)
```

```
END  ExternalizeFields;
```

PROCEDURE **ExternalizeFields** (ptr: ANYPTR; mask: SET; VAR wr: Stores.Writer)
A utility procedure that uses *wr* to Externalize ALL EXPORTED (as read/write) fields of *ptr^* whose type is an element of *mask*.
The only types supported are the simple types BOOLEAN, CHAR, INTEGER, LONGINT, REAL, & SET.
These are coded into *mask* using the type values defined in *Meta*, eg *Meta.boolTyp*.

Pre
20 Attempted to write a field whose type is not an element of LibMisc.simpleFieldTyps.